

# On-the-Flyware

by Diane Boudreau

Software engineers earn their wings by "flying" through development projects

**U**PGRADE NOW!" "NEW FEATURES!" "BETTER PERFORMANCE!" THE ADVERTISEMENTS ARE EVERYWHERE. BUT IS THE SOFTWARE? EVERY DAY, CONSUMERS ARE BOMBARDED WITH ADS PLUGGING RELEASES OF NEW SOFTWARE PRODUCTS. YET, THEY MAY HAVE TO WAIT MONTHS, OR EVEN YEARS, FOR THE PRODUCTS TO HIT THE STORE SHELVES. EVEN AN INDUSTRY GIANT LIKE MICROSOFT ISN'T IMMUNE. AT THE START OF 1995, MILLIONS OF WINDOWS™ USERS AWAITED THE RELEASE OF THE MUCH-TOUTED WINDOWS 95™ OPERATING SYSTEM. BUT THE YEAR WAS HALF OVER BEFORE CUSTOMERS COULD BUY THE PRODUCT. WHAT'S GOING ON? JIM COLLOFELLO CALLS THE PROBLEM THE "SOFTWARE CRISIS."

"Most software projects tend to be late," explains the Arizona State University professor of computer science and engineering. "There are many reasons why projects are late or over budget. Most often, it is because they weren't properly planned. The bigger the project, the more technology risk associated with it, the more likely that delay will occur."

Most software developers and project managers are expert programmers. But developing software requires much more than programming ability. It also requires business and communication skills that are important for managing staff, keeping to a budget, and prioritizing tasks.

These tasks are known collectively as project management. They are integral to any software development project. Unfortunately, computer science graduates often enter the workforce lacking these less-technical skills.

In a typical project management class, students design and create a piece of software as part of a one-semester group project. Such work gives them some experience in the development process, but it does not address real-life issues like staff burnout and attrition.

Collofello is working on a computer program that may help project managers "earn their wings" in a different way. At first glance, the interface looks more like the cockpit of an airplane than a management tutorial—it is covered with gauges and dials. In fact, the Software Development Management (SDM) Simulator was modeled after flight simulators like the ones used for Air Force training.

Created by Collofello and Doug Sycamore, a senior software engineer at Motorola, the program simulates a complete development project in a very short time. The interface was made to look like a flight simulator because, like flying, project management requires an individual to account for many variables.

"You've got to be concerned with a lot of different things," explains Sycamore, who conceived of the project while still a graduate student at ASU. "You could be right on target while you're flying an aircraft. But if you're out of gas, you're in big trouble. You've got to look at all the different gauges and instruments."

Like pilots, project managers also control variables such as staffing,

scheduling, budgeting, and evaluation. Students using the simulator make decisions about these factors that ultimately lead to a project's success or failure. Unlike typical class projects, the simulator does not require students to create an actual program in order to see results. As a result, they can focus on management issues without having to write code.

"What would take months to do is done in minutes here," Collofello says. "We give them immediate feedback. We put them into a lot of very controlled situations and scenarios in a very compressed time-frame. It's much better than simply reading about it or hearing about it or simply encountering it on your own [in the workplace]."

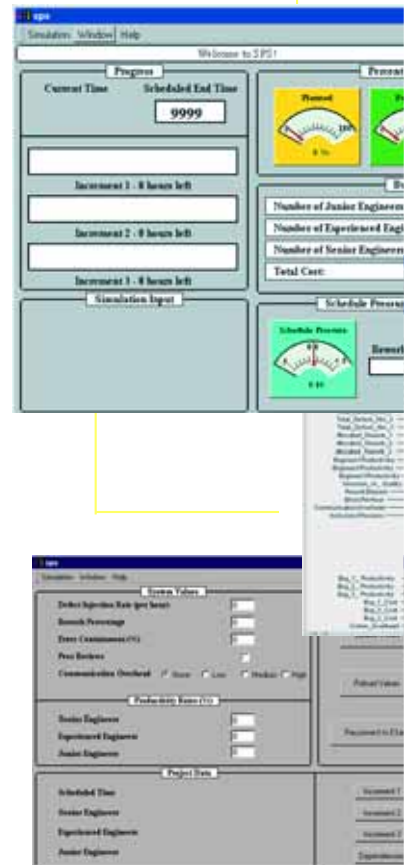
The idea for the SDM Simulator took off in 1994 when Sycamore was a master's student at ASU. He and doctoral student John Tvedt were brainstorming about Sycamore's thesis topic. Their idea was to create a "flight simulator" that used system dynamics modeling for project management.

At the time, Tvedt was developing system dynamics models. He collaborated with Sycamore on much of the initial work. Since earning his degree in 1996, Sycamore has continued to develop the simulator with Collofello.

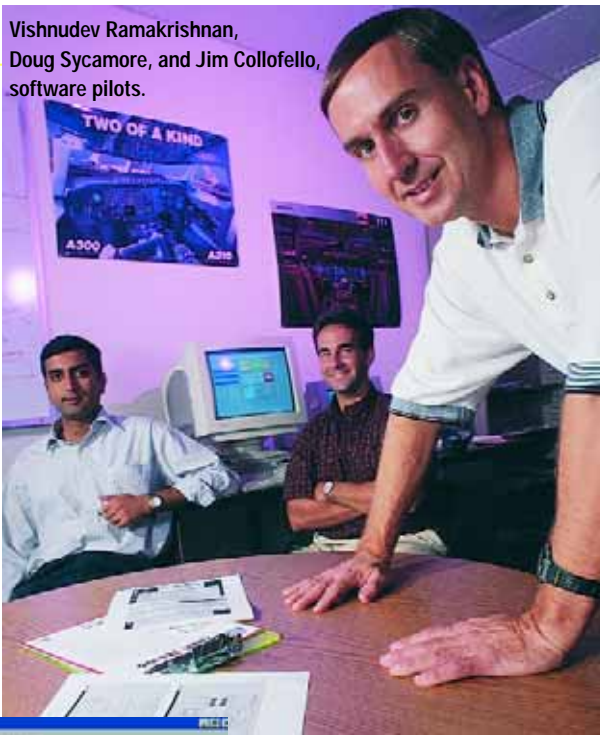
"Doug stuck around after he graduated. He didn't just grab his diploma and run," says Collofello, who uses the SDM Simulator in his graduate project management class. Motorola began using the tool in a Motorola University class in November 1997.

Jim Gerwitz is a section manager for Motorola's Government Space Systems Division. He was one of the students in that class. Gerwitz was intrigued by what he had heard about the simulator, and wanted to try it for himself.

"We give them immediate feedback. We put them into a lot of very controlled situations and scenarios in a very compressed time-frame."



Vishnudev Ramakrishnan,  
Doug Sycamore, and Jim Collofello,  
software pilots.



Courtesy James Collofello, Ph.D. / John C. Phillips photo



often believe that adding more staff to a project will proportionately reduce the time it takes to complete the project. They do not consider the time it takes to train new people, added time spent communicating among the group, or the fact that some tasks can only be done by one person at a time.

“If you add twice as many people to the project you’re not going to cut your schedule in half,” Sycamore says. “If you want to have a baby and you have two women do it, you still won’t get it done in 4 1/2 months!”

Another common phenomenon is schedule pressure. When a project falls behind schedule, employees often work harder and longer to compensate—but only to a point. After time, workers reach a burnout phase. They actually become less productive. If the pressure continues, employees start handing in their resignations.

“All these different dynamic variables affect a project. You can’t just linearly or statistically do it one sequence after another,” Sycamore says. “You must consider all these feedbacks, all these elements that are affecting the project.”

Using the SDM Simulator, students get a shot at managing a development project from start to finish. They explore concepts such as the mythical man-month and schedule pressure. They learn how to set up staffing, scheduling, and evaluation. They also learn about the effectiveness of peer reviews and defect prevention.

“The program can tell you things that weren’t intuitively obvious, or even something that you could figure out,” Gerwitz adds. “People often got an answer they hadn’t really expected.”

As part of their training, Gerwitz and his classmates were given a hypothetical project with a specific budget, deadline, and labor pool. The students assigned staff members to different phases of the project. They also made decisions about scheduling, evaluations, and other management issues. The simulator graphed the progress of the project according to schedule, quality, percent complete, and budget.

Gerwitz says that most of the students in his class did pretty well. The new tool helped them fine-tune their skills. “After running our original scenario, we went back through and asked, ‘How can we change this?’ We were able to optimize the strategy and reduce the hours,” he says.

How exactly does the computer “know” what the results of a scenario will be?

While designing the simulator, Sycamore and Collofello gathered data from Motorola and other industry records. They also spent considerable time talking to colleagues in the field. However, they both stress that the simulator was not designed to be generic to all companies.

“From a pure academic point of view, we want to use industry standard types of data,” Collofello says. “However, to be useful for a particular company, you really need to calibrate the program with data from that company. Industry average is useless for any particular company.”

In the future, Sycamore plans to move the simulator beyond the classroom and into the office as a tool to plan and track actual software projects.

“That’s what is good about this project,” he says. “I feel like I’ve contributed to the computer science community. Many master’s theses sit on a shelf collecting dust. No one ever sees the work. I’ve benefited from the work. Hopefully, others will as well.”

“I think everybody was really impressed with the tool. I know I was,” Gerwitz says. “The simulator is very applicable to our situation. It fits the way we use resources. It would be very helpful on the job.”

There are other project-management tools on the market that help with scheduling, budget tracking, and time reporting. This tool’s foundation in system dynamics makes it special.

Sycamore defines system dynamics as the cause and effect feedback loops that have non-linear influences on a system. In other words, every part of the system affects every other part. And any actor may be affected by its own action.

Development projects are full of these interrelationships, Sycamore explains. “As you build your staffing profile, the experience level of your staff is going to affect your productivity. That’s going to affect your schedule and your effort.

“Your experience level is also going to affect the defects that are injected into the system. That’s going to affect your rework hours. If you staff your whole project with junior engineers, you’re going to have a lot higher rework hours.”

These relationships seem like common sense, but not all of them are. One misconception is the “mythical man-month.” Project managers

*The SDM Simulator project is supported by Motorola. For more information, contact Jim Collofello, Ph.D., Department of Computer Science and Engineering, 480.965.3190. Send e-mail to: collofello@asu.edu.*